<https://learn.upgrad.com/course/983/segment/8101/52582/151558/814568>

# **Introduction**

Welcome to the module on ‘Principal Component Analysis’.

****Principal component analysis**** (PCA) is one of the most commonly used dimensionality reduction techniques in the industry. By converting large data sets into smaller ones containing fewer variables, it helps in improving model performance, visualising complex data sets, and in many more areas.

## In this module

Let's hear from your SME Mirza Rahim Baig as he introduces the topic of PCA

Play

Mute

Current Time 0:00

/

Duration 0:39

Loaded: 0%

2x

Playback Rate

Quality Levels

Picture-in-PictureFullscreen

1569002

Now let's get to know what you'll be studying in this module and what are the necessary pre-requisites for the same.

As explained in the aforementioned video, the  entire module has been divided into the following main sections:

* ****Fundamentals of PCA****: Here, you will get an idea of why you should learn about PCA and its essential building blocks before understanding the process. This has been divided into 2 sub-sessions:
  + Fundamentals of PCA I
  + Fundamentals of PCA II
* ****PCA Using Python****:  Here, you will implement PCA using Python and get to know its various applications.

## Prerequisites

This module requires prior knowledge of certain linear algebra concepts, such as matrices, vectors, etc. You will get to know about those prerequisites, along with a brief overview of each, as you go through the sessions. You can also learn the same from the additional module on ‘[Maths for Data Analysis](https://learn.upgrad.com/v/course/983/module/44629" \t "https://learn.upgrad.com/course/983/segment/8101/52582/151558/_blank)’, which contains some useful additional content and questions to improve your understanding of these concepts. Here is a checklist of the concepts that you need to know to understand this module:

* Vectors and their properties
* Vector operations (addition, scaling, linear combination and dot product)
* Matrices
* Matrix operations (matrix multiplication and matrix inverses)

## In this session

First, in order to fully appreciate PCA’s usefulness, you will look at a wide variety of situations - some of which you may have encountered in your earlier modules, like the multicollinearity problem and how PCA helps us solve it. Then, you will learn the basic definition of PCA, followed by a brief introduction to linear algebra topics that are crucial for understanding PCA and its building blocks. After this, you will look at two key ideas that form the workings of PCA: change of ****basis**** and ****variance****as information.

## Guidelines for in-module questions

The in-video and in-content questions for this module are not graded. The graded questions are given in a separate segment labelled 'Graded Questions' at the end of this session****.**** The questions in that segment will adhere to the following guidelines:

|  |  |  |
| --- | --- | --- |
|  | First Attempt Marks | Second Attempt Marks |
| Question  with 2 Attempts | 10 | 5 |
| Question  with 1 Attempt | 10 | 0 |

## 

## People you'll hear from in this module

[Mirza Rahim Baig](https://www.linkedin.com/in/mirza-rahim-baig-27b4a316/" \t "https://learn.upgrad.com/course/983/segment/8101/52582/151558/_blank)  
Analytics Lead at Flipkart

Seasoned advanced analytics/ data science professional with 10 years of experience in advanced analytics, machine learning, consulting in the e-commerce and healthcare domains. Proficient in machine learning and applications; adept at solving complex problems through data.

# **The Why of PCA**

The first thing to know before learning anything new is to understand why and how that knowledge is useful. Hence, let's start by understanding the motivation for studying PCA and then look at a brief overview of the technique and its applications.

****Note****: At 3:05, for 100 variables we will need ****4950****plots to visualise the associations, not 450.

Play Video

1569002

As explained by Rahim, a couple of situations where having a lot of features posed problems for us are as follows:

* The predictive model setup: Having a lot of correlated features lead to the multicollinearity problem. Iteratively removing features is time-consuming and also leads to some information loss.
* Data visualisation: It is not possible to visualise more than two variables at the same time using any 2-D plot. Therefore, finding relationships between the observations in a data set having several variables through visualisation is quite difficult.

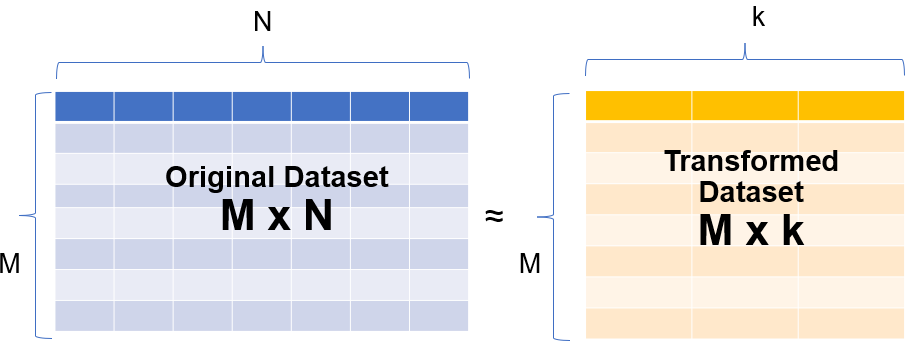
Now, PCA helps in solving both the problems mentioned above which you'll study shortly.

Let’s listen to the following lecture to understand the various applications of PCA.

Play Video

1569002

Fundamentally, PCA is a dimensionality reduction technique, i.e., it approximates the original data set to a smaller one containing fewer dimensions(Notethat ***dimension is just another term for referring to columns or variables in a dataset***). To understand it visually, take a look at the following image.

****

In the image above, you can see that a data set having N dimensions has been approximated to a smaller data set containing 'k' dimensions. In this module, you will learn how this manipulation is done. And this simple manipulation helps in several ways such as follows:

* For data visualisation and EDA
* For creating uncorrelated features that can be input to a prediction model:  With a smaller number of uncorrelated features, the modelling process is faster and more stable as well.
* Finding latent themes in the data: If you have a data set containing the ratings given to different movies by Netflix users, PCA would be able to find latent themes like genre and, consequently, the ratings that users give to a particular genre.
* Noise reduction

Now attempt the following questions to test your understanding.

**Question 1/2**

Mandatory

#### **PCA**

When can (or should) PCA be used?

When the attributes of your data are highly correlated

When better data visualisation is possible using less number of dimensions

When the number of dimensions needs to be reduced

All of the above

**✓ Correct**

**Feedback:**

Yes, that's correct!

**Your answer is Correct.**

**Attempt 1 of 2**

**Question 2/2**

Mandatory

#### **Multicollinearity**

Which of the following is ****not true**** about multicollinearity?

Multicollinearity generally occurs when there are high correlations between two or more predictor variables.

In the case of multicollinearity, some of the predictor variables can be used to predict some other predictor variables.

An easy way to detect multicollinearity is to calculate correlation coefficients for all combinations of predictor variables.

**✕ Incorrect**

**Feedback:**

This is correct.

Multicollinearity helps in better convergence of a regression problem.

**✓ Correct**

**Feedback:**

Multicollinearity leads to incorrect estimates of the model parameters. Hence, never helps in a better convergence.

**Your answer is Wrong.**

**Attempt 2 of 2**

# **The What of PCA**

As discussed in the previous segment, PCA is fundamentally a ****dimensionality reduction technique****; it helps in manipulating a data set to one with fewer variables. The following lecture will give you a brief idea of what dimensionality reduction is and how PCA helps in achieving dimensionality reduction.

Play Video

1569002

In simple terms, dimensionality reduction is the exercise of dropping the unnecessary variables, i.e., the ones that add no useful information. Now, this is something that you must have done in the previous modules. In EDA, you dropped columns that had a lot of nulls or duplicate values, and so on. In linear and logistic regression, you dropped columns based on their p-values and VIF scores in the feature elimination step.

Similarly, what PCA does is that it converts the data ****by creating new features from old ones****, where it becomes easier to decide which features to consider and which not to.

Now that you have an idea of the basics of what PCA does, let’s understand its definition in the following lecture.

Play Video

1569002

PCA is a statistical procedure to convert observations of possibly correlated variables to ‘principal components’ such that:

* They are ****uncorrelated**** with each other.
* They are ****linear combinations**** of the original variables.
* They help in capturing maximum ****information**** in the data set.

Now, the aforementioned definition introduces some new terms, such as ‘****linear combinations****’ and ‘****capturing maximum information****’, for which you will need some knowledge of linear algebra concepts as well as other building blocks of PCA. In the next session, we will start our journey in the same direction with the introduction of a very basic idea: the ****vectorial representation of data****.

Answer the following question to better understand the upcoming segments.

**Question 1/1**

Mandatory

#### **PCA**

Which of the following is/ are true regarding PCA (principal component analysis)?

It is an unsupervised technique

Principal components are the linear combinations of original variables

Principal components are constructed to capture the maximum information

All of the above

**✓ Correct**

**Feedback:**

Go through the definition of PCA.

**Your answer is Correct.**

**Attempt 1 of 2**

# **Summary**

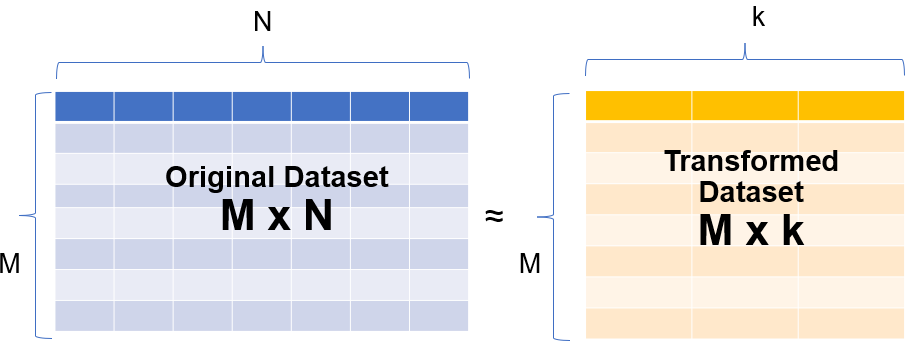
Here's a brief summary of what you learnt in this session!

Dimensionality reduction is a way of transforming a dataset having a high number of features into a smaller dataset. A couple of situations where having a lot of features posed problems for us are as follows:

* The predictive model setup: Having a lot of correlated features lead to the multicollinearity problem. Iteratively removing features is time-consuming and also leads to some information loss.
* Data visualisation: It is not possible to visualise more than two variables at the same time using any 2-D plot. Therefore, finding relationships between the observations in a data set having several variables through visualisation is quite difficult.

In simple terms, dimensionality reduction is the exercise of dropping the unnecessary variables, i.e., the ones that add no useful information. Now, this is something that you must have done in the previous modules. In EDA, you dropped columns that had a lot of nulls or duplicate values, and so on. In linear and logistic regression, you dropped columns based on their p-values and VIF scores in the feature elimination step.

 PCA is one such dimensionality reduction technique, i.e., it approximates the original data set to a smaller one containing fewer dimensions. What PCA does is that it converts the data ****by creating new features from old ones****, where it becomes easier to decide which features to consider and which not to.  To understand it visually, take a look at the following image.

****

In the image above, you can see that a data set having N dimensions has been approximated to a smaller data set containing 'k' dimensions. In this module, you will learn how this manipulation is done. And this simple manipulation helps in several ways such as follows:

* For data visualisation and EDA
* For creating uncorrelated features that can be input to a prediction model:  With a smaller number of uncorrelated features, the modelling process is faster and more stable as well.
* Finding latent themes in the data: If you have a data set containing the ratings given to different movies by Netflix users, PCA would be able to find latent themes like genre and, consequently, the ratings that users give to a particular genre.
* Noise reduction

As explained in the video above, PCA is a statistical procedure to convert observations of possibly correlated variables to ‘principal components’ such that:

* They are ****uncorrelated**** with each other.
* They are ****linear combinations**** of the original variables.
* They help in capturing maximum ****information**** in the data set.

**Question 1/1**

Mandatory

#### **Top 3 Takeaways**

What are you top 3 takeaways from this session?

Word Count **30**Word Limit **5 - 100**

**Attempt 1 of 1**

# **Graded Questions**

**Question 1/1**

Mandatory

#### **PCA - Properties**

Consider the following statements

Statement 1 - PCA helps in solving the multicollinearity problem by creating new uncorrelated features which are used as input for the predictive model.

Statement 2 - With a dimensionality reduction technique like PCA, you convert a dataset having N dimensions to another dataset having k dimensions where N > k.

Now choose the correct option.

Only Statement 1 is correct

Only Statement 2 is correct

Both the statements are correct

**✓ Correct**

**Feedback:**

PCA does create uncorrelated features which solve the problem of multicollinearity. PCA also reduces the number of dimensions from N to k (or N > k )

None of the statements are correct.

**✕ Incorrect**

**Feedback:**

Statement 1 -  What's the basic cause of multicollinearity and how does PCA resolve it?

Statement 2 - Do dimensions get reduced or increased in case of dimensionality reduction?

**Your answer is Wrong.**

**Attempt 2 of 2**

# **Introduction**

Welcome to the session on Fundamentals of PCA - I!

Here we'll learn about two of the most important building blocks of PCA - ****basis****and ****change of basis****. But before that, we'll go through a brief refresher on basic linear algebra concepts.

## In this session

This session covers some important linear algebra concepts required for understanding PCA and how it works. These main concepts are as follows

* Vectors
* Matrices and their Inverse
* Basis vectors
* Change of Basis
* PCA and Change of Basis

## Guidelines for in-module questions

The in-video and in-content questions for this module are not graded. The graded questions are given in a separate segment labelled 'Graded Questions' at the end of the session. The questions in that segment will adhere to the following guidelines:

|  |  |  |
| --- | --- | --- |
|  | First Attempt Marks | Second Attempt Marks |
| Question  with 2 Attempts | 10 | 5 |
| Question  with 1 Attempt | 10 | 0 |

## 

## People you'll hear from in this module

[Mirza Rahim Baig](https://www.linkedin.com/in/mirza-rahim-baig-27b4a316/" \t "https://learn.upgrad.com/course/983/segment/8101/52582/151559/_blank)  
Analytics Lead at Flipkart

Seasoned advanced analytics/ data science professional with 10 years of experience in advanced analytics, machine learning, consulting in the e-commerce and healthcare domains. Proficient in machine learning and applications; adept at solving complex problems through data.

## Coming Up

In the next segment, you will be taken through Linear Algebra topics required to know in order to learn PCA.

# **Vectorial Representation of Data**

In order to understand the workings of PCA, it is crucial to understand some essential linear algebra concepts, such as matrices, vectors and their associated operations. Let’s take a look at the following lecture as you go through a checklist of linear algebra that you should be knowing before foraying into PCA.

Play Video

1569002

To summarise what you're going to learn in this segment here's a handy checklist:

* Vectors and their properties
* Vector operations (addition, scaling, linear combination and dot product)
* Matrices
* Matrix operations (matrix multiplication and matrix inverses)

Let's start with understanding the dataset as a matrix of vectors in the following lecture.

Play Video

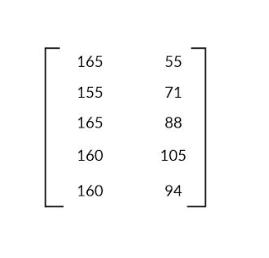
1569002

Note - In the video at 1:48 the graphic mistakenly shows [16565] instead of [16555]

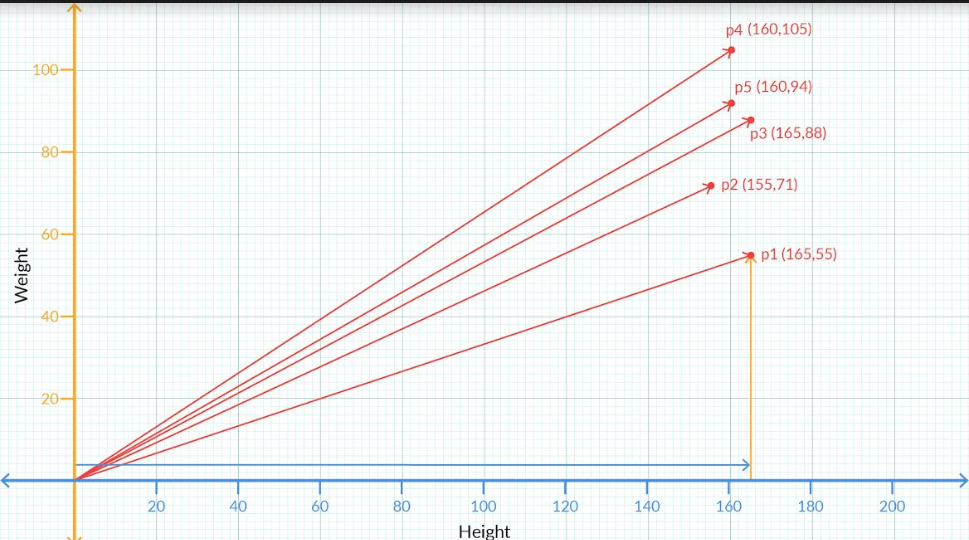
As mentioned in the video, consider the following data set containing the height and weight of five patients.



The height and weight information can be represented in the form of a matrix as follows

****

with each row representing a particular patient's data and each column representing the original variable. Geometrically, these patients can be represented as shown in the following image.

****

[****Note****: The point P5 is slightly off from its actual position in the graph given above and in the video.]

## ****Vector Representation****

The vector associated with the first patient is given by the values (165, 55). This value can also be written in the following way:

1. 1. A column containing the values along the rows. This is also known as the column-vector representation.  
   [16555]
2. As a transpose of the above form. Essentially, it is the same column vector but now written as a transpose of a row vector.  
   [16555]T  
   [Note: Transpose is something you must have learnt in your Python for DS  module. If you need some brushing up on this topic, you can take a look at this[link](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.transpose.html" \t "https://learn.upgrad.com/course/983/segment/8101/52582/151559/_blank)]
3. In terms of the basis vectors   
   This is something that you'll learn in detail in later segments. To give a brief idea, the vector (165,55) can also be written as 165****i**** +55****j****, where ****i**** and ****j**** are the unit vectors along X and Y respectively and are the basis vectors used to represent all vectors in the 2-D space.

## ****Vector Representation for n-dimensional data****

Each vector will contain values representing all the dimensions or variables in the data. For example, if there was an age variable also included in the above dataset and the first patient had an age of 22 years, then the vector representing him would be written as  (165, 55, 22). Similarly, if the dataset had 10 variables, there would be 10 dimensions in the vector representation. Similarly, you can extend it for n dimensions or variables.

Now, these vectors have certain properties and operations associated with them. Let's go ahead and learn them in the next segment. Before that, you can attempt the following question to test your understanding until now.

**Question 1/1**

Mandatory

#### **Vector Representation**

The following matrix shows the coordinates of a couple of locations: L1 and L2.

⎡⎢⎣LocationXYL112.314.7L218.519.6⎤⎥⎦

What is the vectorial representation of the Location L2?

(18.5, 19.6)

[18.519.6]

[18.519.6]T

All the above options are correct.

**✓ Correct**

**Feedback:**

The vector can be represented in all the ways as mentioned above.

**Your answer is Correct.**

**Attempt 2 of 2**

# **Vector Operations**

Now that you've understood what vectors are, let's go ahead and learn about some vector properties and a few associated operations.

Play Video

1569002

Let's summarise the learnings of the above lecture. Major outcomes from the above video are:

1. ****Vectors have a direction and magnitude****  
   Each vector has a direction and magnitude associated with it. The direction is given by an arrow starting from the origin and pointing towards the vector's position. The magnitude is given by taking a sum of squares of all the coordinates of that vector and then taking its square root.  
     
   For example, the vector (2,3) has the direction given by the arrow joining (0,0) and (2,3) pointing towards (2,3). Its magnitude is given by  √22+32=√13.  
     
   Similarly, for a vector in 3 dimensions, say (2,-3,4) its direction is given by the arrow joining (0,0,0) and (2,-3,4) pointing towards (2,-3,4). And as in the 2D case, we get the magnitude of this vector as  √(2)2+(−3)2+(4)2=√29 .
2. ****Vector Addition****  
   When you add two or more vectors, we essentially add their corresponding values element-wise. The first element of both the vectors get added, the second element of the both get added and so on.  
   For example, if you've two vectors say   
   V1=(2,3) and V2=(1,2) then   
   V1+V2=(2+1,3+2)=(3,5).
3. In the****i, j****notations that we introduced earlier, the above addition can be written as V1+V2=(2i+3j)+(i+2j)=(2+1)i+(3+2)j=3i+5j  
   Similarly, this idea can be extended to multiple dimensions as well.
4. ****Scalar Multiplication****  
   If you multiply any real number or scalar by a vector, then there is a change in the magnitude of the vector and the direction remains same or turns completely opposite depending on whether the value is positive or negative respectively.

**Question 1/2**

Mandatory

#### **Unit Vectors**

Unit vectors are those vectors that have a unit magnitude and are signifiers of a particular direction. To find a unit vector along the direction of another vector, you divide that vector by its magnitude.

For example, if there is a vector A=Axi+Ayj whose magnitude is M, then a unit vector along the direction of A is given by AxMi+AyMj

What is the unit vector along the direction of the vector 3i + 4j?

0.8i - 0.6j

0.8i + 0.6j

0.6i - 0.8j

0.6i + 0.8j

**✓ Correct**

**Feedback:**

The magnitude of the vector is given by √(32+42)=√25=5). Therefore a unit vector along A is given by (3/5)****i**** + (4/5)****j****= 0.6****i****+0.8****j****

**Your answer is Correct.**

**Attempt 1 of 2**

Continue

Coming Up

In the next video, you will learn how to perform multiplication operation on matrices.

**Question 1/2**

Mandatory

#### **Unit Vectors**

Unit vectors are those vectors that have a unit magnitude and are signifiers of a particular direction. To find a unit vector along the direction of another vector, you divide that vector by its magnitude.

For example, if there is a vector A=Axi+Ayj whose magnitude is M, then a unit vector along the direction of A is given by AxMi+AyMj

What is the unit vector along the direction of the vector 3i + 4j?

0.8i - 0.6j

0.8i + 0.6j

0.6i - 0.8j

0.6i + 0.8j

**✓ Correct**

**Feedback:**

The magnitude of the vector is given by √(32+42)=√25=5). Therefore a unit vector along A is given by (3/5)****i**** + (4/5)****j****= 0.6****i****+0.8****j****

**Your answer is Correct.**

**Attempt 1 of 2**

**Question 2/2**

Mandatory

#### **Vector Addition**

Let v1 and v2 be two vectors given by v1=i+2j and v2=2i−3j. Find the value of 2⋅v1+3⋅v2.

2i+3j

5i+4j

8i−5j

**✓ Correct**

**Feedback:**

2\*v1 +  3\*v2 = 2(i + 2j) + 3(2i -3 j) = 2i +4j + 6i - 9j = 8i -5j

4i+j

**Your answer is Correct.**

**Attempt 1 of 2**

**Question 2/2**

Mandatory

#### **Vector Addition**

Let v1 and v2 be two vectors given by v1=i+2j and v2=2i−3j. Find the value of 2⋅v1+3⋅v2.

2i+3j

5i+4j

8i−5j

**✓ Correct**

**Feedback:**

2\*v1 +  3\*v2 = 2(i + 2j) + 3(2i -3 j) = 2i +4j + 6i - 9j = 8i -5j

4i+j

**Your answer is Correct.**

**Attempt 1 of 2**

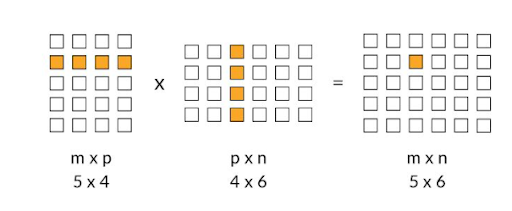
# **Matrix Multiplication**

Apart from the vector operations that we learnt previously, we need some knowledge of matrix operations as well. Let's hear from Mirza as he explains the idea of matrix multiplication.

Play Video

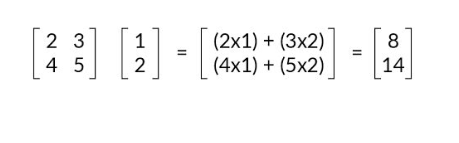
1569002

As you saw in the video, the process of matrix multiplication is quite simple, and it involves element-wise multiplication followed by the addition of all the elements present in it. The one key rule that it must satisfy is when you multiply 2 matrices, say A and B, the number of columns of A must equal the number of rows in B. Visually, you can take a look at the following image to get the idea of how that should be.



As shown in the example, since the number of columns in the first matrix and the number of rows in the second matrix are equal to 4, matrix multiplication is possible and the resultant matrix has a shape of 5 x 6.

The element-wise multiplication followed by addition is also pretty straightforward as can be seen in the following example.



Here is a short video that shows how to do matrix multiplication in Python.

Play Video

1569002

Now answer the following questions to make sure that you've understood this concept in detail.

**Question 1/2**

Mandatory

#### **Matrix Multiplication**

If matrix dimensions of two matrices are given, say,  A =a1 x a2 (dimensions) and  B=b1 x b2 (dimensions), then matrix multiplication A\*B  is valid if \_\_?

a1=b1

a1=b2

a2=b1

**✓ Correct**

**Feedback:**

We need the number of columns of A, i.e a2, to be equal to the number of rows of B, i.e b1, for matrix multiplication to be possible.

a2=b2

**Your answer is Correct.**

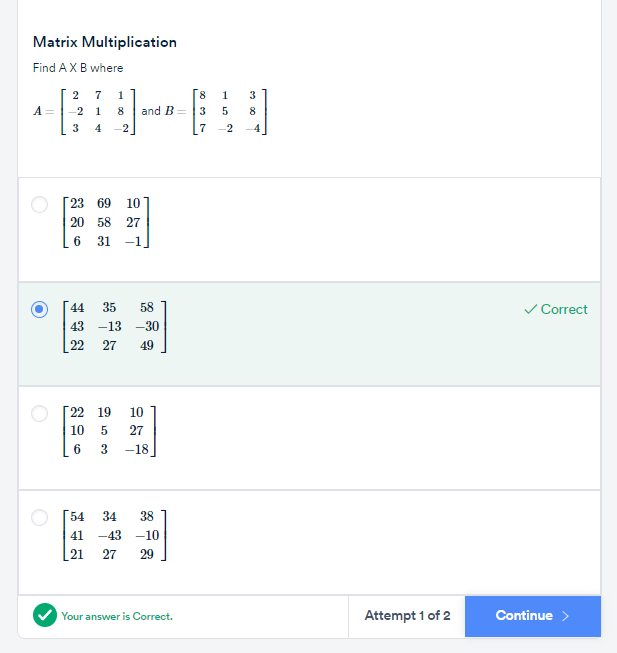
**Attempt 1 of 2**

**Question 2/2**

Mandatory

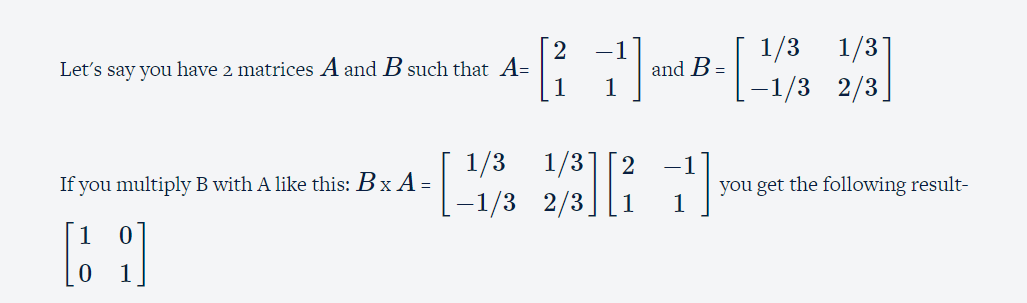
#### **Matrix Multiplication**

**Attempt 1 of 2**



## ****Inverse of a Matrix****

To understand what the inverse of a matrix is, let's take a look at the following example:



The matrix that you got after the multiplication above is also known as an ****Identity matrix****. In matrix notation, it serves the same function as that of the number 1 in the real number system. To establish an analogy, in the real number system if you multiply any number by 1, you get the number itself. Similarly, when you multiply any matrix with the identity matrix, also denoted by I, you get the same matrix once again. ( You can calculate this and verify yourselves)

Now taking the analogy of the real number system, when you multiply 2 numbers  a and b and it comes out to be 1, i.e.

a×b=1

then a and b are called reciprocal of each other.

In the matrix world, if you have two matrices A and B, and their multiplication results in the identity matrix  I, i.e.

B  x A = I,

then A and B are called ****inverses**** of each other.

The inverse of A is also written as A−1.  Therefore B=A−1

In a later segment, we'll get to know how these inverses are useful.

Note that A−1A=I=AA−1. You can verify this using the above matrix.

Here's a short video showing how to find the inverse of a matrix in Python.

Play Video

1569002

Now, use Numpy to calculate the inverse of the matrix in the following question.

**Question 1/1**

Mandatory

#### **Matrix Inverse**

## 

## Additional Reading

* If you want to learn how to find the inverse of a Matrix mathematically, please refer to this [link](https://www.khanacademy.org/math/algebra-home/alg-matrices/alg-intro-to-matrix-inverses/v/inverse-matrix-part-1" \t "https://learn.upgrad.com/course/983/segment/8101/52582/151559/_blank).

In the next segment, you will learn how we express the vectors of a matrix.

# **Basis**

In the previous segments, you have learnt how to represent vectors and matrices and understood some of their important operations. We will dive into one of the most fundamental building blocks of PCA:****Basis****. But before we get into the math part of it, let’s understand, in a very intuitive way, what it represents, in the following lecture.

Play Video

1569002

Essentially, ‘basis’ is a unit in which we express the vectors of a matrix.

For example, we describe the weight of an object in terms of the kilogram, gram, and so on; to describe length, we use a metre, centimetre, etc. So for example, when you say that an object has a length of 23 cm, what you are essentially saying is that the object’s length is 23×1 cm. Here, 1 cm is the unit in which you are expressing the length of the object.

Similarly, vectors in any dimensional space or matrix can be represented as a linear combination of basis vectors.

Let’s discuss them in further detail in the following lecture.

Play Video

1569002

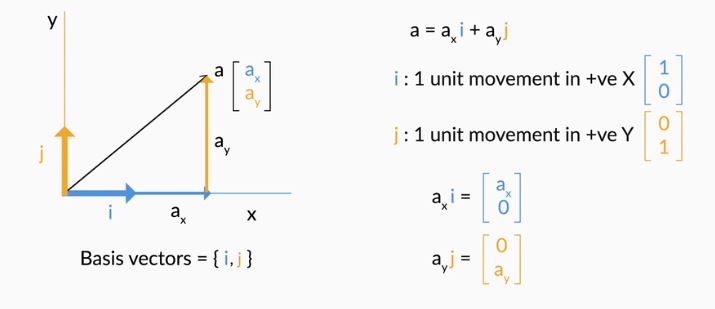
Let's unpack the ideas that you learnt in the above video. Since i and j themselves represent ****(1,0)**** and ****(0,1)****, you can represent any vector in the 2-D space with these i and j vectors.

 Any vector ****'****a****'**** (ax,ay) can be represented in a 2-D space, using the following notation:

a=axi+ayj   
or

a=ax⋅[10]+ay⋅[01]

Visually, it can be represented as follows:



For example, a vector A (2,3) can be written as 2⋅[10]+3⋅[01]. In order to obtain the vector A, we scaled ****i**** by 2 and****j**** by 3 and then finally added them up.

This scaling and adding the vectors up to obtain a new vector is also known as a ****linear combination****.

For the patients' dataset that we had earlier, we can denote each patient vector by the following notation.

IMG_257

Therefore, now we can say that Patient 1 is represented by 165(1 cm,0) + 55(0,1kg). And similarly, we can express other patients' information as well.

The ****basic definition**** of basis vectors is that they're a certain set of vectors whose linear combination is able to explain any other vector in that space.

In a 2D space, the standard basis vectors are given by [10] and [01]. In a 3D space, the same are given by ⎡⎢⎣100⎤⎥⎦, ⎡⎢⎣010⎤⎥⎦ and ⎡⎢⎣001⎤⎥⎦. As you can see, an n-dimensional space or a dataset having n variables would have n standard basis vectors.

In the next segment you will learn understand how you can use different basis to explain the same set of vectors.

**Question 1/1**

Mandatory

#### **Basis Vectors**

The vectors i and j are the basis of two-dimensional space. Which of the following is true?

The magnitude of both these vectors is unity

The linear combination of i and j can represent any point on the two-dimensional space

****i**** cannot be expressed in terms of****j**** and vice versa

All of the above

**✓ Correct**

**Feedback:**

i =(1,0) and j=(0,1) are unit vectors with magnitudes of 1.

Any point on the 2-D space is a linear combination of i and j.

i and j are orthogonal/ perpendicular vectors, and hence, one can't be represented by the other.

**Your answer is Correct.**

**Attempt 1 of 2**

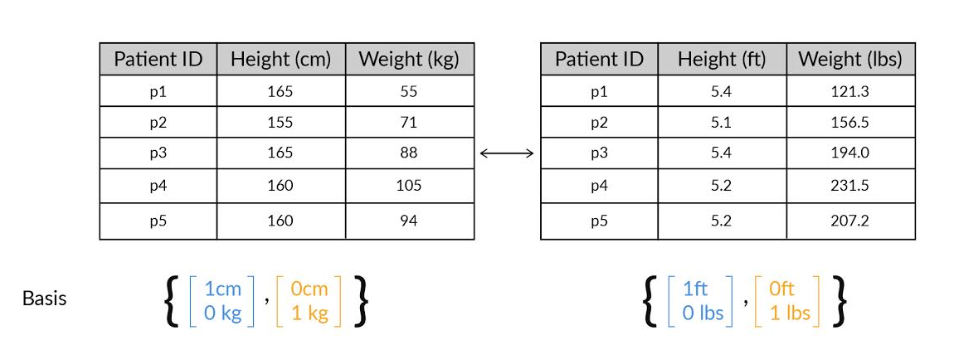
# **Change of Basis: Introduction**

In the previous segment, you understood the concept of basis vectors and how they're the most fundamental units through which you explain the vectors. Now, let's go ahead and understand how you can use different basis to explain the same set of vectors, similar to how you can use different units to explain the same measure.

Play Video

1569002

As explained in the video above, using the analogy of basis as a unit of representation, different basis vectors can be used to represent the same observations, just like you can represent the weight of a patient in kilograms or pounds. As in the previous case, the basis vectors for the representation of the patient’s information is given by [1ft0lbs]and [0ft1lbs].  
The following table summarises the results you get when you make the change.



As you can see, the patient's height and weight have not changed physically. It's just that you're using a different set of basis vectors now to explain the same patients. So [16555] is the same as [5.4121.3]when different basis vectors are being used.

## ****Relationship between the two sets of  basis vectors****

To understand the relationship between the two basis vectors in concrete terms, recall the way we introduced it in the previous segment. We said that every vector in the 2D space can be written as a linear combination of the basis vectors.

So Patient 1's information in the cm/kg space is given by 165⋅[10]+55⋅[01] whereas in the ft/lbs space is given by 5.4⋅[10]+121.3⋅[01]

Now, 1 ft = 30.48 cm and 1 cm = 0.033 ft

Similarly, 1 kg = 2.205 lbs and 1lbs = 0.454 kg.

Therefore, comparing the basis vectors, we can say

[1ft0lbs]​in ft/lbs space = [30.48cm0kg] in cm/kg space and

[0ft1lbs] in ft/lbs space = ​[0cm0.45kg]​

Here's a neat manipulation that you can do to understand the way the numbers arrange amongst themselves using the linear combination property.

[16555]  = ****165****[10] + ****55****[01] = ****5.4****[30.480] +****121.3****[00.45]in the cm/kg space.

In the above case, we considered the new basis vectors as [30.480] and [00.45] in the cm/kg space which is equivalent to (1,0) and (0,1) in the ft/lbs space. And using this, we got the representation of [5.4121.3]for the patient.

Therefore, we can choose a completely different set of vectors, say ****v1**** and ****v2**** as the basis vectors and find the representation of  Patient 1 (originally in the standard basis vectors) in the new basis system. They should be satisfying the following linear combination equation

[16555]= a1⋅v1+a2⋅v2

where (a1,a2) is the representation of Patient 1 in the v1 and v2 space.

To understand better,

Taking v1=[30.480] and v2=[00.45] we got a1=5.4 and a2=121.3

Similarly, taking v1=[550] and v2=[055] we get a1=3 and a2=1

Again, taking v1=[31] and v2=[20] we get a1=55 and a2=0

and so on..

Did you notice something different in the above example where we considered  v1=[31] and v2=[20]? This means that the new basis need not be parallel to the original basis.

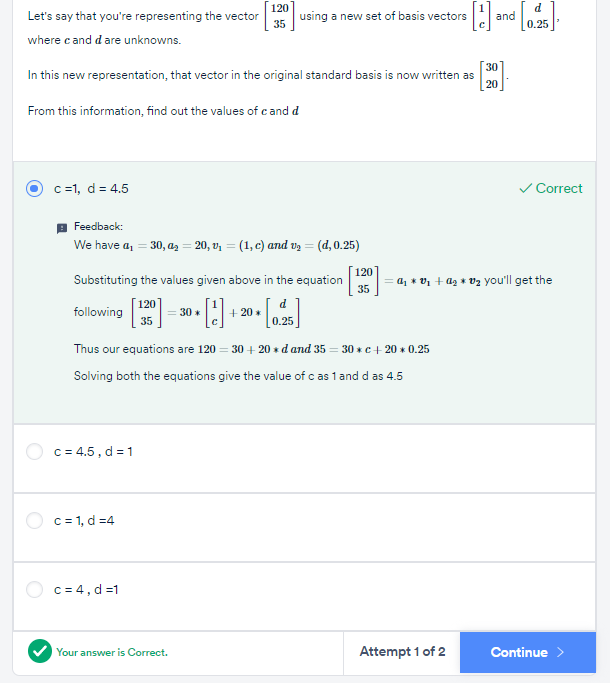
Simply put, you have the flexibility of choosing a different set of basis vectors apart from the standard basis vectors that are provided to you to represent your information. The information won't change, just the numbers representing the information would change.

In the next segment you will do all the necessary calculations that come along in change of basis.

**Question 1/1**

Mandatory

#### **Different Basis Vectors**



# **Introduction**

Welcome to the session on Fundamentals of PCA - II!

Here we'll learn about the concept of variance and it's importance and use in the PCA, concept of basis vectors and how you can use different basis vectors to represent the same information.

## In this session

This session covers some important linear algebra concepts required for understanding PCA and how it works. These main concepts are as follows

* Introduction to Variance
* Variance as Information
* Directions of Maximum Variance
* The Workings of PCA

## People you'll hear from in this module

[Mirza Rahim Baig](https://www.linkedin.com/in/mirza-rahim-baig-27b4a316/" \t "https://learn.upgrad.com/course/983/segment/8101/52582/151560/_blank)  
Analytics Lead at Flipkart

Seasoned advanced analytics/ data science professional with 10 years of experience in advanced analytics, machine learning, consulting in the e-commerce and healthcare domains. Proficient in machine learning and applications; adept at solving complex problems through data.

# **Introduction to Variance**

In the previous session, you learnt about the first fundamental building block for learning PCA - the idea of basis and the change of basis. You saw how a simple change of basis led to dimensionality reduction in the case of the roadmap example and then understood how you can represent the same data in multiple basis vectors.

However, we didn't know how to find those "ideal basis vectors" and what exact properties they must satisfy. In this session, we'll get to do that by understanding the idea of ****variance as information****.

Play Video

1569002

As mentioned previously, you have already learnt certain methods through which you delete columns – by checking the number of null values, unnecessary information and in modelling by checking the p-values and VIF scores.

PCA gauges the importance of a column by another metric called ‘****variance****’ or how varied a column’s values are.

Let's go ahead and look at some examples in the next segment and get an intuitive idea of what variance actually means.

# **Variance as Information**

Let's take a look at a simple example that will help us intuitively understand how variance in the data is equivalent to information we can extract out of the data.

Play Video

1569002

As you saw in the example, the first image didn't have much information in it. Speaking of it in the ways the pixels are arranged, it is the same colour throughout. However, there is a lot of things that you could distinguish easily in the second image and therefore that image has a lot to offer in terms of information. The pixels have a lot of variety and therefore that image has more variance and equivalently, more information.

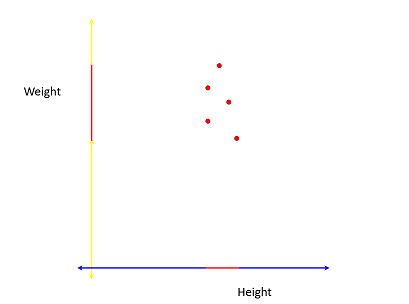
Play Video

1569002

So the key takeaway from the above lecture is to measure the importance of a column by checking its variance values. If a column has more variance, then this column will contain more information.

****Geometrical Interpretation of  Variance****

In the above example, you saw that the variance of height was only 14, whereas that of weight was 311.14. This gave you an idea that Weight is a more important column than Height. Now, there is another elegant way of looking at variance geometrically. Take a look at the following image.



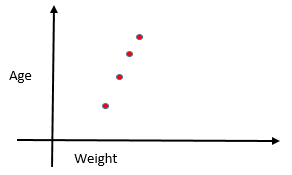
The red line on the Height and Weight axes show the****spread**** of the projections of the vectors on those axes. As you can see here, the spread of the line is quite good on the Weight axis as compared to the Height axis. Hence you can say that Weight has more variance than Height. This idea of the spread of the data being equivalent to the variance is quite an elegant way to distinguish the important directions from the non-important ones.

**Question 1/1**

Mandatory

#### **PCA: Variance**

Which axis captures more variance in the following plot between Age and Weight?



Age

**✓ Correct**

**Feedback:**

When the farthest points of the data are projected on the axis, the length of the projection becomes proportional to the variance. In the given image, the length of the projection of the farthest points of data on the Y-axis is more than the length of the projection on the X-axis.

Weight

**✕ Incorrect**

**Feedback:**

Check if the spread is higher for Weight Axis or not.

Both have equal variances

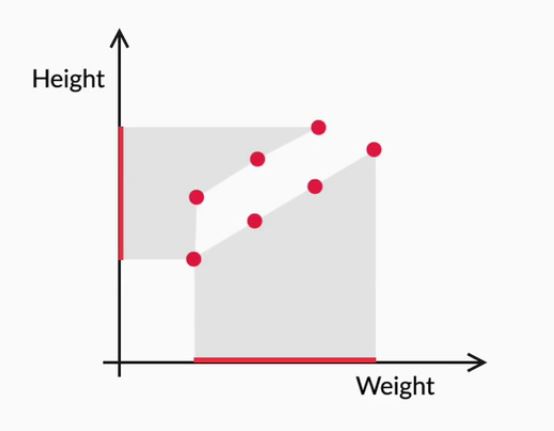
**Your answer is Wrong.**

**Attempt 2 of 2**

# **Directions of Maximum Variance**

So you saw that when the variances are unequally distributed among the original features or columns i.e. some columns have much less variance than others, it is easier to remove those columns and do dimensionality reduction.

But what about the scenario when the variances are pretty similar? For example, take a look at the following image containing the height and weight information of a different group of patients.



As you can see, the spread along both the axes is quite comparable and therefore, you can't directly go and say that one direction is more useful than the other. What to do now?

Let’s look at the next lecture to further understand this problem and appreciate how PCA solves this problem smartly.

Play Video

1569002

After going through the above lecture, you have more or less understood what PCA does. It changes the basis vectors in such a way that the new basis vectors capture the maximum variance or information. In the next video, we'll get to know how this happens visually.

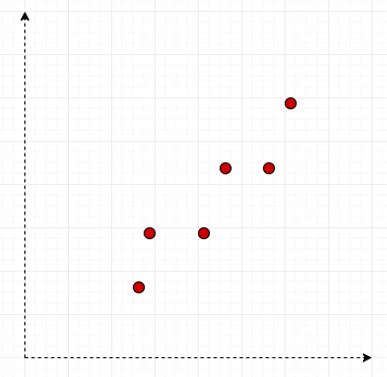
Play Video

1569002

Basically, the steps of PCA for finding the principal components can be summarised as follows.

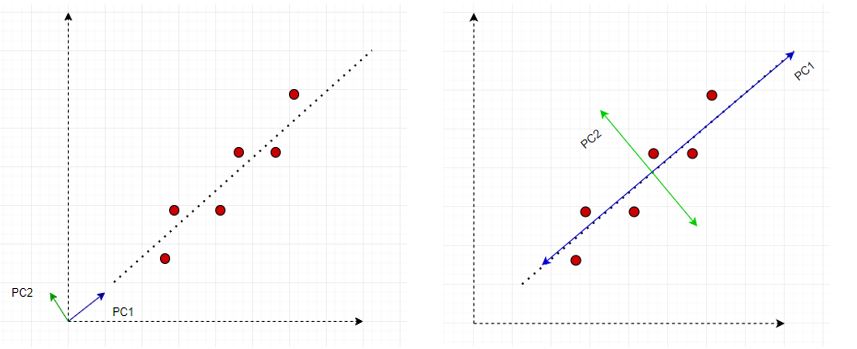
* First, it finds the basis vector which is along the best- fit line that maximises the variance. This is our first ****principal component or PC1.****
* The second principal component is perpendicular to the first principal component and contains the next highest amount of variance in the dataset.
* This process continues iteratively, i.e. each new principal component is perpendicular to all the previous principal components and should explain the next highest amount of variance.
* If the dataset contains *****n***** independent features, then PCA will create *****n***** Principal components.

For a 2-D dataset that has the representation as shown in the image below.



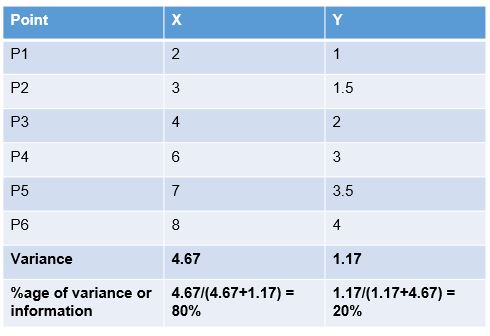
 The principal components can be visually represented as shown in the image below.

(Click on the image to enlarge it.)

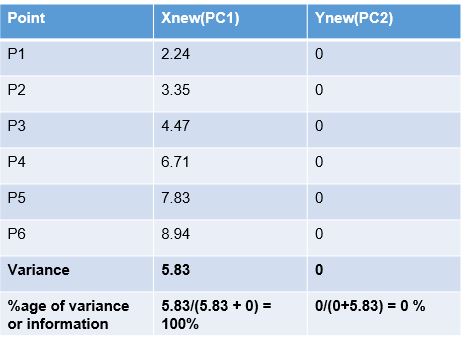


Also, once the Principal Components are found out, PCA assigns a %age variance to each PC. Essentially it's the fraction of the total variance of the dataset explained by a particular PC. This helps in understanding which Principal Component is more important than the other and by how much. This is shown in the images below.

****Original Dataset****

****

****PCA Modified Dataset****



Since 100% of the total variance or information of the entire dataset is present in only one of the columns (PC1) we can safely drop PC2 and still be assured of losing no information.

In the next video you will learn about the objectives that PCA aims to achieve.

# **The Workings of PCA**

Until now, you've learnt the two building blocks of PCA: Basis and variance. In the following video, we will make use of both the terms to make you understand the objective that PCA aims to achieve.

Play Video

1569002

The steps  of PCA as summarised in the above video are as follows:

* Find n new features - Choose a different set of n basis vectors (non-standard). These basis vectors are essentially the directions of maximum variance and are called Principal Components
* Express the original dataset using these new features
* Transform the dataset from the original basis to this PCA basis.
* Perform dimensionality reduction - Choose only a certain k (where k < n) number of the PCs to represent the data.  Remove those PCs which have fewer variance (explain less information) than others.

PCA's role in the ML pipeline almost solely exists as a dimensionality reduction tool. Basically, you choose a fixed number of PCs that explained a certain threshold of variance that you have chosen and then uses only that many columns to represent the original dataset. This modified dataset is then passed on to the ML pipeline for further prediction algorithms to take place. PCA helps us in improving the model performance significantly and helps us in visualising higher-dimensional datasets as well.

****Additional Reading****

* As mentioned in the video, you can take a look at the[Algorithm of PCA optional session](https://learn.upgrad.com/v/course/983/session/147373/segment/794190" \t "https://learn.upgrad.com/course/983/segment/8101/52582/151560/_blank)to understand in detail about how PCA finds the new basis vectors using the eigendecomposition of the covariance matrix method.

# **Summary: II**

Let's reiterate the learnings of the past two sessions:

* You understood the concept of basis vectors and how they're helpful in the representation of data points.
* You then learnt about how you can use different basis vectors to represent the same information.
* Using the previous knowledge, you came to know that when represented under some '****ideal basis vectors'****, it becomes easier for us to do dimensionality reduction. However, you didn't exactly know how to find those ideal basis vectors.
* Then you learnt about the concept of variance and how more variance meant more information.
* Then you derived that the more important columns in a dataset are the ones which capture more variance than the others.
* Subsequently, you deduced that the most important directions, rather than just columns, are those that ****capture maximum variance****. The ideal basis vectors that we talked about in the previous case are in fact those that do the same.
* These basis vectors or directions that capture the maximum variance are essentially the ****Principal Components****for the dataset.

**Question 1/1**

Mandatory

#### **Top 3 Takeaways**

What are your top 3 takeaways from this session?

Word Count **33**Word Limit **5 - 100**

**Attempt 1 of 1**

# **Introduction**

Welcome to the final session of PCA. In the previous session, you discovered and learnt the theoretical concepts of PCA. In this session, you will learn how to implement PCA in python on some real examples.

In this session, you will learn how to use PCA on a problem you have already encountered before - predicting telecom churn using logistic regression. You will now learn to implement PCA in tandem with logistic regression.

## In this session

Let's look at the broad flow of this session.

Play Video

1569002

## Prerequisites

There are no prerequisites for this session other than the knowledge of the previous sessions.

## Guidelines for in-module questions

The in-video and in-content questions for this module are not graded. The graded questions are given in a separate segment labelled 'Graded Questions' at the end of the session. The questions in that segment will adhere to the following guidelines:

|  |  |  |
| --- | --- | --- |
|  | First Attempt Marks | Second Attempt Marks |
| Question  with 2 Attempts | 10 | 5 |
| Question  with 1 Attempt | 10 | 0 |

## People you will hear from in this session

****Subject Matter Expert****

[Mirza Rahim Baig](https://www.linkedin.com/in/rahim-baig/" \t "https://learn.upgrad.com/course/983/segment/8101/52582/151561/_blank)

Analytics Lead, Flipkart

Seasoned advanced analytics/ data science professional with 10 years of experience in advanced analytics, machine learning, consulting in the e-commerce and healthcare domains. Proficient in machine learning and applications; adept at solving complex problems through data.

# **Applying PCA using Python**

For this demonstration, we begin with a very popular machine learning dataset - 'Iris'. In the next few segments, you will learn the necessary steps needed to perform PCA on a dataset and then appreciate how it helps in visualising your data that contains more than two dimensions.

You can download the dataset and the python notebook used in the demonstration from the link given below:

**PCA : Demonstration**

**Download**

Play Video

1569002

Here is a summary of the important steps that you've performed in the video and something that you should do whenever you perform PCA on any other dataset as well.

1. After basic data cleaning procedures, standardise your data

2. Once standardisation has been done, you can go ahead and perform PCA on the dataset. For doing this you import the necessary libraries from sklearn.decomposition.

**from** **sklearn.decomposition** **import** PCA

3. Instantiate the PCA function and set the random state to some specific number so that you get the same result every time you execute that code. (If you want to learn more about random state and how it works, you can check this [StackOverflow answer](https://stackoverflow.com/a/42197534" \t "https://learn.upgrad.com/course/983/segment/8101/52582/151561/_blank))

pca = PCA(random\_state=**42**)

4.  Perform PCA on the dataset by using the pca.fit function.

pca.fit(x)

5. The Principal Components can be accessed using the following code:

pca.components\_

Executing the above code will give the list of Principal components of the original dataset. They'll be of the same number as the original variables in your dataset. In the next segment, you shall see how to choose the optimal number of principal components.

# **Scree Plots**

In the previous segment, you learnt how to perform PCA on your dataset and obtain the Principal Components. The final PCs that you got were as follows:

array([[ **0.52237162**, -**0.26335492**, **0.58125401**, **0.56561105**],

[ **0.37231836**, **0.92555649**, **0.02109478**, **0.06541577**],

[-**0.72101681**, **0.24203288**, **0.14089226**, **0.6338014** ],

[-**0.26199559**, **0.12413481**, **0.80115427**, -**0.52354627**]])

PC1 is given by the direction - [0.52  -0.26  0.58   0.56], PC2 by  [0.37 0.92 0.02 0.06] and so on. The principal components of the same number as that of the original variables with each Principal Component explaining some amount of variance of the entire dataset. This information would enable us to know which Principal Components to keep and which to discard to perform Dimensionality Reduction.

Let's understand it further in the following demonstration, where you'll also come to know about ****scree plots**** and how they help in communicating the variance information very effectively.

Play Video

1569002

Here's a summary of the important steps that you performed :

1. First, you came to know how much variance is being explained by each Principal Component using the following code:

pca.explained\_variance\_ratio\_

The values that you got were as follows:

array([**0.72770452**, **0.23030523**, **0.03683832**, **0.00515193**])

The above values can be summarised in the following table:

|  |  |
| --- | --- |
| Principal  Component | Variance explained  (in %) |
| PC1 | 72.8 |
| PC2 | 23 |
| PC3 | 3.6 |
| PC4 | 0.5 |

So as you can see, the first PC, i.e. Principal Component 1([0.52  -0.26  0.58   0.56]) explains the maximum information in the dataset followed by PC2 at 23% and PC3 at 3.6%. In general, when you perform PCA, all the Principal Components are formed in decreasing order of the information that they explain. Therefore, the first principal component will always explain the highest variance, followed by the second principal component and so on. This order helps us in our dimensionality reduction exercise, as now we know which directions are more important than the others.

Now, in our dataset, we only had 4 columns and equivalently 4 PCs. Therefore it was easy to visualise the amount of variance explained by them using a simple bar plot and then we're able to make a call as to how much variance to keep in the data. For example, using the table above, you only need 2 principal components or 2 directions (PC1 and PC2) to explain more than 95% of the variation in the data.

But what happens when there are hundreds of columns? Using the above process would be cumbersome since you'd need to look at all the PCs and keep adding their variances up to find the total variance captured.

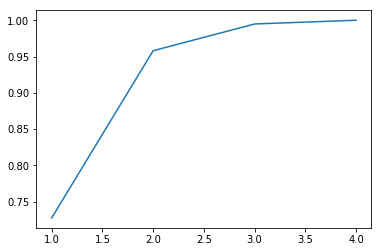
2. Using a ****Scree-Plot****

An elegant solution here would be to simply add a plot of "Cumulative variance explained chart". Here against each number of components, we have the total variance explained by all the components till then.

|  |  |  |
| --- | --- | --- |
| Principal  Component | Variance Explained(in %) | Cumulative Variance Explained (in %) |
| PC1 | 72.8 | 72.8 |
| PC2 | 23 | 95.8 |
| PC3 | 3.6 | 99.4 |
| PC4 | .5 | 99.9 |

So for example, cumulative variance explained by the top 2 principal components is the sum of their individual variances, given by 72.8 +23 =95.8 %. Similarly, you can continue this for 3 and 4 components.

If you plot the number of components on the X-axis and the total variance explained on the Y-axis, the resultant plot is also known as a Scree-Plot. It would look somewhat like this:



Now, this is a better representation of variance and the number of components needed to explain that much variance.

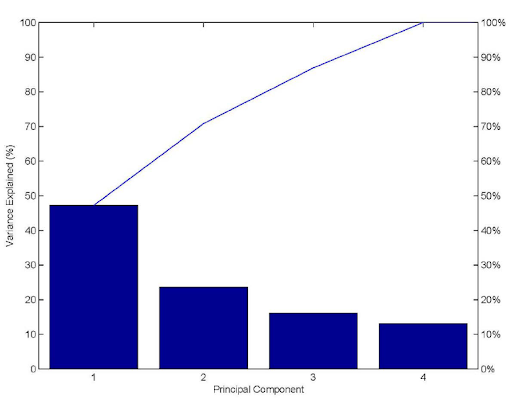
Answer the following questions:

**Question 1/1**

Mandatory

#### **Scree- Plots**

In the following scree plot, how much total information is retained using only three principal components?



More than 95%

More than 80%

**✓ Correct**

**Feedback:**

Note that the fourth component explains just above 10% variance but less than 20% variance.  Therefore, the rest of the three components explain more than 80% variance.

Exactly 60%

Exactly 70%

**Your answer is Correct.**

**Attempt 1 of 2**

# **Dimensionality Reduction**

In the previous two segments, you understood how to apply PCA on a dataset followed by the importance of scree-plots. Now that you know how many principal components you need to explain a certain amount of variance, let's go and finally do dimensionality reduction on our dataset using the Principal Components that we've chosen.

Play Video

1569002

Here's a summary of the important steps that you saw above:

1.) Choosing the required number of components

From the scree plot that you saw previously, you decided to keep ~95% of the information in the data that we have and for that, you need only 2 components. Hence you instantiate a new PCA function with the number of components as 2. This function will perform the dimensionality reduction on our dataset and reduce the number of columns from 4 to 2.

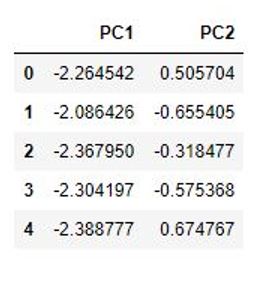
pc2 = PCA(n\_components=**2**, random\_state=**42**)

2.) Perform ****Dimensionality Reduction**** on our dataset.

Now you simply transform the original dataset to the new one where the columns are given by the Principal Components. Here you've finally performed the dimensionality reduction on the dataset by reducing the number of columns from 4 to 2 and still retain 95% of the information. The code that you used to perform the same step is as follows:

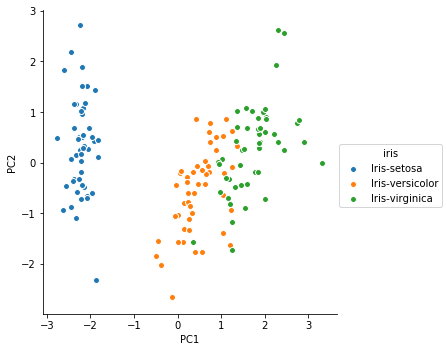
newdata = pc2.fit\_transform(x)

and the new dataset is given as follows:



3) Data Visualisation using the PCs

Now that you have got the data in 2 dimensions, it is easier for you to visualise the same using a scatterplot or some other chart. By plotting the observations that we have and dividing them on the basis of the species that they belong to we got the following chart:



As you can see, you clearly see that all the species are well segregated from each other and there is little overlap between them. This is quite good as such insight was not possible with higher dimensions as you won't be able to plot them on a 2-D surface. So, therefore, applying  PCA on our data is quite beneficial for observing the relationship between the data points quite elegantly.

****Important Note****: When you perform PCA on datasets generally, you may need more than 2 components to explain an adequate amount of variance in the data. In those cases, if you want to visualise the relationship between the observations, choose the top 2 Principal Components as your X and Y axes to plot a scatterplot or any such plot to do the same. Since PC1 and PC2 explain the most variance in the dataset, you'll be getting a good representation of the data when you visualise your dataset on those 2 columns.

# **Practice Questions: I**

Here's a brief summary of what you've learnt so far:

* Applying PCA on a dataset in python
* Evaluate the amount of variance explained by each component
* Use the scree-plot to choose how much variance you need to explain with your transformed dataset
* Transform the dataset to the new chosen Principal Components and then perform dimensionality reduction
* Use the new dataset for visualisation of the observations

Now here are some questions to test your understanding of the same.

**Question 1/3**

Mandatory

#### **PCA in Python**

The pca.fit() function does which of the following tasks?

Only explains the variance in the dataset

Performs PCA on the dataset and finds the Principal Componets.

**✓ Correct**

**Feedback:**

pca.fit() is used to perform PCA on the dataset.

Only plots the scree-plot for the components

None of the above.

**Your answer is Correct.**

**Attempt 2 of 2**

**Question 2/3**

Mandatory

#### **PCA in Python**

Let's say a friend of yours performed PCA on a dataset containing 10 columns and decided to keep only 3 components for the final transformed data. For finding the final transformed data, he used the following code. Evaluate the code and choose the right option.

pca = PCA(random\_state = **42**)

newdata = pca.fit\_transform(dataset)

The given code is correct and would result in getting the transformed dataset in 3 dimensions.

The code has syntax issues and hence would throw up an error

The random\_state parameter should be 56 instead of 42

**✕ Incorrect**

**Feedback:**

The random\_state only makes sure that the code result is reproducible every time you run it.

Another parameter n\_components = 3 should be present in the pca function

**✓ Correct**

**Feedback:**

The function should be

pca = PCA(n\_components =**3** ,random\_state=**42**

**Your answer is Wrong.**

**Attempt 2 of 2**

**Question 3/3**

Mandatory

#### **PCA in Python**

For data visualisation purposes after performing PCA, mostly PC1 and PC2 are chosen because

They have the highest variances  explained along their directions.

**✓ Correct**

**Feedback:**

Since PC1 and PC2 have the highest and the second highest variances explained along their directions, they'll be able to summarise the dataset more effectively than any other pair of components.

They are orthonormal

There is no specific reason.

They are uncorrelated.

**Your answer is Correct.**

**Attempt 1 of 2**

# **Improving Model Performance - I**

In the previous segments, you saw how to perform dimensionality reduction using PCA and then immediately were introduced to one of its key applications which are for data visualisation. However, the most common application of PCA is to improve your model's performance. So in real life, you use PCA in conjunction with any other model like Linear Regression, Logistic Regression, Clustering amongst others in order to make the process more efficient. In the following demonstration, you'll be looking at both the scenarios - performing model building without PCA and then with PCA to appreciate how much faster it is to get similar or better results in the latter case.

Download the datasets and the python notebook for this demonstration from the link given below.

**Model Building with PCA**

**Download**

[****Note**** - At a few places, the codes have deprecated in the latest python libraries. Kindly make the changes as shown below

* Line 11 - *telecom['TotalCharges']=telecom['TotalCharges'].convert\_objects(convert\_numeric=True)*

                Please change it to :

          -  *telecom['TotalCharges']=pd.to\_numeric(telecom['TotalCharges'],errors='coerce')*

* Line 37

          - *y\_pred\_final = y\_pred\_final.reindex\_axis(['CustID','Churn','Churn\_Prob'], axis=1)*

                Please change it to:

          - *y\_pred\_final = y\_pred\_final.reindex(['CustID','Churn','Churn\_Prob'], axis=1)*

****Overview of the Demo****

For this demonstration, our main model will be a logistic regression setup. As mentioned above, first we'll be performing Logistic Regression directly without any PCA. For this demo, we'll be using the Telecom Churn dataset that you have worked earlier with.

****Model Building without PCA****

Since you're already familiar with the data and the logistic regression model that you built, here's a quick walkthrough to refresh your memory.

Play Video

1569002

****Video Correction:****At 03:52, Rahim says 'linear regression' though he meant 'logistic regression'.

In the video below, we will apply PCA on the data and visualise it by the transformations done by the PCA.

Play Video

1569002

You saw the process of building a churn prediction model using logistic regression. Some important problems with this process that Rahim pointed out are:

* ****Multicollinearity**** among a large number of variables, which is not totally avoided even after reducing variables using RFE (or a similar technique)
* Need to use a ****lengthy iterative procedure****, i.e. identifying collinear variables, using variable selection techniques, dropping insignificant ones etc.
* A ****potential loss of information****due to dropping variables
* ****Model instability**** due to multicollinearity

If you remember the first session, we discussed all these points as potential issues that plague our model building activity. Now let's go ahead and perform PCA on the dataset and then apply Logistic Regression and see if we get any better results.

****Model Building with PCA****

In the second part, first, we'll reduce the dimensions that we have using PCA and then create a logistic regression model on it.

As you could see, with PCA, you could achieve the same results with just a couple of lines of code. It will be helpful to note that the baseline PCA model has performed at par with the best Logistic Regression model built after the feature elimination and other steps.

PCA helped us solve the problem of multicollinearity (and thus model instability), loss of information due to the dropping of variables, and we don't need to use iterative feature selection procedures. Also,  our model becomes much faster because it has to run on a smaller dataset. And even then, our ROC score, which is a key model performance metric is similar to what we achieved previously.

To sum it up, if you're doing any sort of modelling activity on a large dataset containing lots of variables, it is a good practice to perform PCA on that dataset first, reduce the dimensionality and then go ahead and create the model that you wanted to make in the first place. You are advised to perform PCA on the datasets that you worked on in Linear Regression and Clustering as well, to see how it makes our job easier.

In the next segment you will learn about another functionality in PCA where you can select the amount of variance that you want your final dataset to capture.

# **Improving Model Performance - II**

Till now, you've been looking at the scree-plot to choose the number of components that explain a certain amount of variance before going for the dimensionality reduction using PCA. Now, there is a nice functionality which makes this process even more unsupervised. All you need to do is select the amount of variance that you want your final dataset to capture and PCA does the rest for you. Let's take a look at the following demonstration to see how we can do the same.

Play Video

1569002

As you saw above, all you needed to do was select a particular amount of variance that you want to be explained by the Principal Components of the transformed dataset. PCA automatically chooses the appropriate number of components on its own and proceeds with the transformation. This again saves us a lot of time!

In the next segment we have provided practise questions, answer them based on the concepts learnt in this session.

# **Practice Questions: II**

Here's a summary of your learnings in the last two segments:

* You understood the importance of PCA in model building. Essentially before you build any model, you perform PCA to reduce its dimensionality.
* This results in a smaller dataset with uncorrelated features - thereby leading to faster execution and a much more stable model.
* You observed that performing PCA and then doing the actual model greatly improves its efficiency and also does that without any iterative procedures.

Now answer the following questions by applying PCA on the given dataset. Note that you don't need to perform any scaling operation here. Just directly perform PCA.

**PCA**

**Download**

**Question 1/2**

Mandatory

#### **PCA in Python**

Which function in sklearn.decomposition do you use to project the data to the new Principal Components after you've performed the PCA?

pca

pca.components\_

pca.fit\_transform()

**✓ Correct**

pca.fit

**Your answer is Correct.**

**Attempt 2 of 2**

Continue

**Question 2/2**

Mandatory

#### **PCA in Python**

Once you perform PCA on the dataset provided in this segment and project the data, what is the approximate variance explained by the first principal component?

31%

40%

**✕ Incorrect**

**Feedback:**

Use pca.explained\_variance\_ratio to find the variance explained by all the components.

80%

65%

**✓ Correct**

**Feedback:**

Use pca.explained\_variance\_ratio to find the variance explained by all the components. You can clearly see that the first principal component explains about 65% of the variance.

**Your answer is Wrong.**

**Attempt 2 of 2**

In the next segment you will look at some practical considerations that need to be kept in mind while applying PCA.

# **Practical Considerations and Alternatives**

Until now, you know the in and out of PCA and how to implement it in Python, and hence, you should be aware of when to apply PCA. Let's now look at some practical considerations that need to be kept in mind while applying PCA.

Play Video

1569002

Those were some important points to remember while using PCA. To summarise:

* Most software packages use SVD to compute the principal components and assume that the data is ****scaled and centred,****so it is important to do standardisation/normalisation.
* PCA is a****linear transformation method**** and works well in tandem with linear models such as linear regression, logistic regression, etc., though it can be used for computational efficiency with non-linear models as well.
* It should ****not be used forcefully to reduce dimensionality****(when the features are not correlated).

In the next short lecture, Rahim will talk about some shortcomings of PCA.

Play Video

1569002

You learnt some important shortcomings of PCA:

* PCA is limited to linearity, though we can use ****non-linear techniques such as t-SNE****as well (you can read more about t-SNE in the optional reading material below).
* PCA needs the components to be perpendicular, though in some cases, that may not be the best solution. The alternative technique is to use ****Independent Components Analysis.****
* PCA assumes that columns with low variance are not useful, which might not be true in prediction setups (especially classification problem with a high class imbalance).

If you are interested in reading about t-SNE (t-Distributed Stochastic Neighbor Embedding) or ICA, you can go through the additional reading provided below.

This brings us to the end of this segment.

## ****Additional Reading****

****t-SNE****

* [Laurens van der Maaten's (creator of t-SNE) website](https://lvdmaaten.github.io/tsne/" \t "https://learn.upgrad.com/course/983/segment/8101/52582/151561/_blank)
* [Visualising data using t-SNE: Journal of Machine Learning Research](http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf" \t "https://learn.upgrad.com/course/983/segment/8101/52582/151561/_blank)
* [How to use t-SNE effectively](https://distill.pub/2016/misread-tsne/" \t "https://learn.upgrad.com/course/983/segment/8101/52582/151561/_blank)

****Independent Components Analysis****

* [Stanford notes on ICA](https://sgfin.github.io/files/notes/CS229_Lecture_Notes.pdf" \t "https://learn.upgrad.com/course/983/segment/8101/52582/151561/_blank) (Check pages 122 -127)

# **Summary**

Here's a summary of what you've learnt so far.

First, you implemented PCA in python on the iris dataset. In that demonstration, you understood the basic steps that you need to follow in PCA - on how to perform PCA, find the Principal Components, choose a particular number of components using the scree-plot, transform your data and then visualise the data.

After that, you saw an implementation where you wanted to improve the model efficiency in a Logistic Regression Setup. Here you were able to see that with PCA, you're able to maintain the same level of efficiency without going through all the iterative feature elimination procedures. You also saw how to perform PCA faster by just giving it how much variance you need to be explained.

Here's a list of useful functions that use after importing the PCA function from sklearn libraries.

* ****pca.fit()**** - Perform PCA on the dataset.
* ****pca.components\_ -****Explains the principal components in the data
* ****pca.explained\_variance\_ratio\_**** - Explains the variance explained by each component
* ****pca.fit(n\_components = k)****- Perform PCA and choose only k components
* ****pca.fit\_transform  -****Transform the data from original basis to PC basis.
* ****pca(var) -****Here 'var' is a number between 0-1. Perform PCA on the dataset and choose the number of components automatically such that the variance explained is (100\*var)%.

Please download the lecture notes for this module from the link below

**Lecture Notes: Principal Component Analysis**

**Download**

**Question 1/1**

Mandatory

#### **Top 3 Takeaways**

What are your top 3 takeaways from this session?

Word Count **26**Word Limit **5 - 100**

**Attempt 1 of 1**